2018

Clustering and Multifacility Location With Constraints via Distance Function Penalty Methods and DC Programming

Mau Nam Nguyen
Portland State University, mau.nam.nguyen@pdx.edu

Thai An Nguyen
Portland State University

Sam Reynolds
Portland State University, ser6@pdx.edu

Tuyen Tran
Portland State University, tuyen2@pdx.edu

CLUSTERING AND MULTIFACILITY LOCATION WITH CONSTRAINTS VIA DISTANCE FUNCTION PENALTY METHODS AND DC PROGRAMMING

N. M. NAM[1], N. T. AN[2] , S. REYNOLDS,[3] and T. TRAN[4].

Abstract. This paper is a continuation of our effort in using mathematical optimization involving DC programming in clustering and multifacility location. We study a penalty method based on distance functions and apply it particularly to a number of problems in clustering and multifacility location in which the centers to be found must lie in some given set constraints. We also provide numerical examples to test our method.

Key words. Clustering, DC programming, Nesterov's smoothing techniques, k-mean algorithm

AMS subject classifications. 49J52, 49J53, 90C31.

1 Introduction

In the current time of "big data", clustering is a very important problem that helps classify data in many fields such as machine learning, pattern recognition, image analysis, data compression, and computer graphics. Given a finite number of data points with a measurement distance, a centroid-based clustering problem seeks a finite number of cluster centers with each data point assigned to the nearest cluster center in a way that a certain measurement distance is minimized.

It is well-known that the $k-$mean algorithm is one of the simplest clustering algorithms, providing an easy way to classify a given data set through a certain number of clusters. However, it possesses certain drawbacks: the $k-$mean algorithm depends heavily on the initial choice of cluster centers; there is no guarantee that the $k-$mean algorithm converges to a global optimal solution; the number of clusters k is an input parameter: an inappropriate choice of k may yield poor results; the results depend heavily on the measurement distance; the algorithm may not be applicable for handling constraints imposed on the cluster centers.

In our recent research, we further the pioneering works by Pham Dinh Tao, Le Thi Hoai An and others from [1, 2] in using the mathematical programming approach for clustering, aiming at providing an alternative to the $k-$mean algorithm and coping with its drawbacks; see [1, 7, 8]. The mathematical programming approach is very promising as opti-

[1]Fariborz Maseeh Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, USA (mau.nam.nguyen@pdx.edu). Research of this author was partly supported by the National Science Foundation under grant DMS-1716057.

[2]Institute of Research and Development, Duy Tan University, Vietnam (thaian2784@gmail.com). Research of this author was supported by the Vietnam National Foundation for Science and Technology Development under grant #101.01-2017.325.

[3]Fariborz Maseeh Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, USA (ser6@pdx.edu).

[4]Fariborz Maseeh Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, USA (tuyen2@pdx.edu).

mization techniques for minimizing nonconvex optimization problems have been of great interest with significant progress over the past few years. In addition, it is possible to use derivative-free methods for initializations in the DCA and enhance the effectiveness of gradient/subgradient-based nonconvex algorithms. Our method using Nesterov's smoothing techniques and the DCA, an algorithm for minimizing differences of convex functions, allows us to solve clustering and multifacility location problems in many different settings involving different norms, bilevel clustering, and set clustering.

The main focus of this paper is on solving a number of clustering and multifacility location problems with constraints. We use a penalty method with squared Euclidean distance functions to convert constrained problems to unconstrained problems. Then appropriate DC decompositions and the DCA are used to minimize the penalized objective functions. In the case where the measurement distance is defined by the Euclidean norm instead of the squared Euclidean norm, we use Nesterov's smoothing techniques for *reducing the nonsmoothness* of the model and for providing a DC decomposition that is favorable for applying the DCA. Our method opens up the possibility of using distance function penalty methods for other problems of DC programming.

The paper is organized as follows. In Section 2, we present basic tools of convex analysis and optimization used throughout the paper. The analysis of a penalty method based on squared distance functions is presented in Section 3. Section 4 is devoted to solving clustering problems with constraints in which the measurement distance is defined by the squared Euclidean norm. In Section 5, we study a new model of clustering with constraints that involves sets. In Section 6, we study clustering problems with constrained and the measurement distance defined by the Euclidean norm. These problems belong to the class of continuous multifacility location problems with constraints. Finally, numerical examples are presented in Section 7.

Throughout the paper, we use $\langle \cdot, \cdot \rangle$ to denote the inner product and use $\| \cdot \|$ to denote the associated Euclidean norm in $\mathbb{R}^d$. For the subset Ω of $\mathbb{R}^d$, the set $\text{conv}(\Omega)$ is the convex hull of Ω, i.e., the smallest convex set in $\mathbb{R}^d$ that contains Ω.

2 Preliminaries

In this section, we present basic tools of analysis and optimization used in the sequel. The readers are referred to [1, 4, 5, 11] for more details.

Let $f\colon \mathbb{R}^d \to (-\infty, \infty]$ be a convex function. An element $v \in \mathbb{R}^d$ is called a *subgradient* of f at $\bar{x} \in \text{dom}\,(f) = \{x \in \mathbb{R}^d \mid f(x) < \infty\}$ if it satisfies

$$\langle v, x - \bar{x} \rangle \le f(x) - f(\bar{x}) \text{ for all } x \in \mathbb{R}^d.$$

The set of all such elements v is called the *subdifferential* of f at $\bar{x}$ and is denoted by $\partial f(\bar{x})$. If $\bar{x} \notin \text{dom}\,(f)$, we set $\partial f(\bar{x}) = \emptyset$. This subdifferential concept possesses many calculus rules that are important for applications. In particular, for a finite number of convex functions

$f_i\colon \mathbb{R}^d \to (-\infty,\infty]$, $i=1,\ldots,m$, we have the following sum rule:

$$\partial(f_1+\cdots+f_m)(\bar{x})=\partial f_1(\bar{x})+\cdots+\partial f_m(\bar{x}) \text{ for all } \bar{x}\in\mathbb{R}^d$$

provided that $\bigcap_{i=1}^m \operatorname{ri}(\operatorname{dom}(f_i))\neq\emptyset$. Here $\operatorname{ri}(\Omega)$ stands for the *relative interior* of Ω; see, e.g, [5, Definition 1.68].

If $f=\max_{i=1,\ldots,m} f_i$, and f_i is continuous at $\bar{x}$ for every $i=1,\ldots,m$, then for any $\bar{x}\in\mathbb{R}^d$ we have the following maximal rule:

$$\partial f(\bar{x})=\operatorname{conv}\big(\bigcup_{i\in I(\bar{x})}\partial f_i(\bar{x})\big), \tag{2.1}$$

where $I(\bar{x})=\{i \mid f_i(\bar{x})=f(\bar{x})\}$.

Given a nonempty closed convex subset Ω of $\mathbb{R}^d$ with $\bar{x}\in\Omega$, the *normal cone* to Ω at $\bar{x}$ is defined by

$$N(\bar{x};\Omega)=\big\{v\in\mathbb{R}^d \;\big|\; \langle v,x-\bar{x}\rangle\le 0 \text{ for all } x\in\Omega\big\}.$$

If $\bar{x}\notin\Omega$, we set $N(\bar{x},\Omega)=\emptyset$. It is well-known that an element $\bar{x}\in\mathbb{R}^d$ is an absolute minimizer of a convex function $f\colon\mathbb{R}^d\to\mathbb{R}$ on Ω if and only if $\bar{x}$ is a local minimizer of f on Ω. Moreover, this happens if and only if the following optimality condition holds: $0\in\partial f(\bar{x})+N(\bar{x};\Omega)$.

Let $\Theta\subset\mathbb{R}^d$ be a nonempty set (not necessarily convex). The *distance function* to Θ is defined by

$$d(x;\Theta)=\inf\big\{\|x-w\| \;\big|\; w\in\Theta\big\},\ x\in\mathbb{R}^d.$$

The *Euclidean project* from $x\in\mathbb{R}^d$ to Θ is the set

$$P(x;\Theta)=\big\{w\in\Theta \;\big|\; d(x;\Theta)=\|x-w\|\big\}.$$

We can show that if Θ is a nonempty closed set, then $P(x;\Theta)$ is nonempty, and it is a singleton if we assume in addition that Θ is convex. We can also show that if Θ is a convex set and $w\in P(x;\Theta)$, then $x-w\in N(w;\Theta)$.

Another tool we will use in the paper is the notion of *Fenchel conjugates*. Let $f\colon\mathbb{R}^d\to\mathbb{R}$ be a function. The Fenchel conjugate of f is defined by

$$f^*(y)=\sup\big\{\langle y,x\rangle-f(x) \;\big|\; x\in\mathbb{R}^d\big\},\ y\in\mathbb{R}^d.$$

Note that $f^*\colon\mathbb{R}^d\to(-\infty,\infty]$ is an extended-real-valued convex function. Suppose further that f is convex, then the *Felchel-Moreau theorem* states that $(f^*)^*=f$. Based on this theorem, we have the following relation between the subgradients of f and its Fenchel conjugate:

$$x\in\partial f^*(y) \iff y\in\partial f(x). \tag{2.2}$$

The notions of subgradients and Fenchel conjugates provide mathematical foundation for the DCA introduced below. Given a function $f\colon\mathbb{R}^d\to\mathbb{R}$ with the *DC decomposition* $f=g-h$, where $g,h\colon\mathbb{R}^d\to\mathbb{R}$ are convex functions, the DCA introduced by Pham Dinh

Algorithm 1 : The DCA

INPUT: $x_1, N \in \mathbb{N}$

for $p = 1, \ldots, N$ **do**

Find $y_p \in \partial h(x_p)$

Find $x_{p+1} \in \partial g^*(y_p)$

end for

OUTPUT: x_{N+1}

Tao described in what follows is a simple but effective algorithm for minimizing the function f; see [12, 13].

For convenience, define the *data matrix* $\mathbf{A} \in \mathbb{R}^{m\times d}$ as the matrix whose i^{th} row is $a^i \in \mathbb{R}^d$ for $i = 1, \ldots, m$. Similarly, we define the *variable matrix* $\mathbf{X} \in \mathbb{R}^{k\times d}$ as the matrix whose ℓ^{th} row is $x^\ell \in \mathbb{R}^d$ for $\ell = 1, \ldots, k$. We equip the linear space $\mathbb{R}^{k\times d}$ with the inner product $\langle \mathbf{X}, \mathbf{Y}\rangle = \text{trace}(\mathbf{X}^T\mathbf{Y})$. Recall that the *Frobenius norm* on $\mathbb{R}^{k\times d}$ is defined by

$$\big\|\mathbf{X}\big\|_F = \langle \mathbf{X}, \mathbf{X}\rangle^{1/2} = \left(\sum_{\ell=1}^{k}\langle x^\ell, x^\ell\rangle\right)^{1/2} = \left(\sum_{\ell=1}^{k}\|x^\ell\|^2\right)^{1/2}.$$

Observe that the square of the Frobenius norm is differentiable with

$$\nabla\|\mathbf{X}\|_F^2 = 2\mathbf{X} \text{ for } \mathbf{X} \in \mathbb{R}^{k\times d}.$$

Let $\Omega^\ell \subset \mathbb{R}^d$ for $l = 1, \ldots, k$ be nonempty closed convex sets and let $\mathbf{\Omega} = \Omega^1 \times \Omega^2 \times \ldots \times \Omega^k$. For $\mathbf{X} \in \mathbb{R}^{k\times d}$, the projection from $\mathbf{X}$ to $\mathbf{\Omega}$ is the matrix $\mathbf{Y}$ whose ℓ^{th} row is $y^\ell = P(x^\ell; \Omega^\ell)$. We thus have

$$[d(\mathbf{X}; \mathbf{\Omega})]^2 = \|\mathbf{X} - \mathbf{Y}\|_F^2 = \sum_{\ell=1}^{k}\|x^\ell - y^\ell\|^2 = \sum_{\ell=1}^{k} d(x^\ell; \Omega^\ell)^2. \tag{2.3}$$

3 A Penalty Method via Distance Functions

In this section, we study a penalty method using distance functions for solving constrained optimization problems and apply them specifically to DC programming. This method is based on the *quadratic penalty method*; see [3, 9]. Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a function and let Ω_i for $i = 1, \ldots, q$ be nonempty closed subsets of $\mathbb{R}^d$ with $\bigcap_{i=1}^q \Omega_i \neq \emptyset$. Consider the optimization problem:

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & x \in \bigcap_{i=1}^q \Omega_i. \end{array} \tag{3.1}$$

Let us first study the relation between this problem and the unconstrained problem given by

$$\min\ f_\lambda(x) = f(x) + \frac{\lambda}{2}\sum_{i=1}^{q}[d(x; \Omega_i)]^2,\ x \in \mathbb{R}^d. \tag{3.2}$$

The theorem below provides a relation between optimal solutions of the constrained optimization problem (3.1) and the unconstrained optimization problem (3.2) obtained by a penalty method based on distance functions. The proof follows [9, Theorem 17.1].

Theorem 3.1 *Consider* (3.1) *in which* $f\colon \mathbb{R}^d \to \mathbb{R}$ *is a l.s.c. function. Suppose that* (3.1) *has an optimal solution. If* $\lim_{n\to\infty} \lambda_n = \infty$ *and* $x_n \in \mathbb{R}^d$ *is an absolute minimizer of the function* f_{λ_n} *defined in* (3.2) *for all* $n \in \mathbb{N}$*, then every subsequential limit of* $\{x_n\}$ *is a solution of* (3.1)*.*

Proof. Let $\bar{x} \in \mathbb{R}^d$ be an optimal solution of (3.1). That means $\bar{x} \in \Omega_i$ for $i = 1, \ldots, q$ and

$$f(\bar{x}) \le f(x) \text{ whenever } x \in \Omega_i \text{ for all } i = 1, \ldots, q.$$

Since $x_n \in \mathbb{R}^d$ is an absolute minimizer of the function f_{λ_n},

$$f_{\lambda_n}(x_n) \le f_{\lambda_n}(\bar{x}).$$

This implies, with the observation that $d(\bar{x}; \Omega_i) = 0$ for $i = 1, \ldots, q$, that

$$f(x_n) + \frac{\lambda_n}{2} \sum_{i=1}^{q} [d(x_n; \Omega_i)]^2 \le f(\bar{x}). \tag{3.3}$$

Then

$$\sum_{i=1}^{q} [d(x_n; \Omega_i)]^2 \le \frac{2}{\lambda_n} \Big(f(\bar{x}) - f(x_n) \Big) \text{ for all } n \in \mathbb{N}.$$

Let $x^* \in \mathbb{R}^d$ be a subsequential limit of $\{x_n\}$. Without loss of generality, we can assume that $\lim_{n\to\infty} x_n = x^*$. By the continuity of the distance function and the lower semicontinuity of f,

$$\sum_{i=1}^{q} [d(x^*; \Omega_i)]^2 = \lim_{n\to\infty} \sum_{i=1}^{q} [d(x_n; \Omega_i)]^2 \le \liminf_{n\to\infty} \frac{2}{\lambda_n} \Big(f(\bar{x}) - f(x_n) \Big) \le 0.$$

It follows that $d(x^*; \Omega_i) = 0$, and so $x^* \in \Omega_i$ for $i = 1, \ldots, q$. In addition, by (3.3) and the lower semicontinity of f we have

$$f(x^*) \le \liminf_{n\to\infty} f(x_n) \le \liminf_{n\to\infty} \Big(f(x_n) + \frac{\lambda_n}{2} \sum_{i=1}^{q} [d(x_n; \Omega_i)]^2 \Big) \le f(\bar{x}).$$

Therefore, x^* is an optimal solution of (3.1). □

Now we discuss a direct consequence of Theorem 3.1 that will be used in the sequel. Let $F\colon \mathbb{R}^{k\times d} \to \mathbb{R}$ be a function and let Ω_i^ℓ for $\ell = 1, \ldots, k$ and $i = 1, \ldots, q$ be nonempty closed subsets of $\mathbb{R}^d$. Consider the problem

$$\begin{array}{ll} \min & F(x^1, \ldots, x^k) \\ \text{subject to} & x^\ell \in \bigcap_{i=1}^{q} \Omega_i^\ell \,, x^\ell \in \mathbb{R}^d \text{ for } \ell = 1, \ldots, k. \end{array} \tag{3.4}$$

We now clarify the relation between this problem and the unconstrained problem given by

$$\begin{array}{ll} \min & F_\lambda(x^1,\ldots,x^k) = F(x^1,\ldots,x^k) + \dfrac{\lambda}{2}\displaystyle\sum_{\ell=1}^{k}\sum_{i=1}^{q}[d(x^\ell;\Omega_i^\ell)]^2 \\ & x^\ell \in \mathbb{R}^d \text{ for } \ell = 1,\ldots,k. \end{array} \tag{3.5}$$

In what follows, we identify $\mathbf{X} = (x^1,\ldots,x^k) \in \mathbb{R}^{k\times d}$ with the matrix $\mathbf{X} \in \mathbb{R}^{k\times d}$, whose ℓ^{th} row is x^ℓ for $\ell = 1,\ldots,k$.

Corollary 3.2 *Consider* (3.4) *in which* $F\colon \mathbb{R}^{k\times d} \to \mathbb{R}$ *is a l.s.c. function. Suppose that* (3.4) *has an optimal solution. If* $\lim_{n\to\infty}\lambda_n = \infty$ *and* $X_n = (x_n^1,\ldots,x_n^k) \in \mathbb{R}^{k\times d}$ *is an absolute minimizer of the function* F_{λ_n}*, then every subsequential limit of* $\{X_n\}$ *is a solution of* (3.4).

Proof. Let $\mathbf{X} = (x^1,\ldots,x^k) \in \mathbb{R}^{k\times d}$ and let $\mathbf{\Omega}_i = \Omega_i^1 \times \ldots \times \Omega_i^k \subset \mathbb{R}^{k\times d}$ for $i = 1,\ldots,q$. Note that

$$\bigcap_{i=1}^{q}\mathbf{\Omega}_i = \bigcap_{i=1}^{q}\prod_{\ell=1}^{k}\Omega_i^\ell = \prod_{\ell=1}^{k}\bigcap_{i=1}^{q}\Omega_i^\ell.$$

It follows that $x^\ell \in \bigcap_{i=1}^q \Omega_i^\ell$ for $\ell = 1,\ldots,k$ if and only if $\mathbf{X} \in \bigcap_{i=1}^q \mathbf{\Omega}_i$, and thus (3.4) reduces to the following optimization problem:

$$\begin{array}{ll} \min & F(\mathbf{X}) \\ \text{subject to} & \mathbf{X} \in \bigcap_{i=1}^q \mathbf{\Omega}_i. \end{array}$$

Based on (2.3), we can rewrite the objective function F_λ in (3.5) as follows

$$F_\lambda(\mathbf{X}) = F(\mathbf{X}) + \frac{\lambda}{2}\sum_{i=1}^{q}[d(\mathbf{X};\mathbf{\Omega}_i)]^2.$$

The conclusion now follows directly from Theorem 3.1. □

Let us continue with a known result on DC decompositions of squared distance functions. The proof of the following result can be found in [8, Proposition 5.1].

Proposition 3.3 *Let* Ω *be a nonempty closed set in* $\mathbb{R}^d$ *(not necessarily convex). Define the function*

$$\varphi_\Omega(x) = \sup\big\{\langle 2x, w\rangle - \|w\|^2 \;\big|\; w \in \Omega\big\} = 2\sup\big\{\langle x, w\rangle - \frac{1}{2}\|w\|^2 \;\big|\; w \in \Omega\big\}.$$

Then we have the following conclusions:
(i) *The function* φ_Ω *is always convex. If we assume in addition that* Ω *is convex, then* φ_Ω *is differentiable with* $\nabla\varphi_\Omega(x) = 2P(x;\Omega)$.
(ii) *The function* $f(x) = [d(x;\Omega)]^2$ *is a DC function with* $f(x) = \|x\|^2 - \varphi_\Omega(x)$ *for all* $x \in \mathbb{R}^d$.

We now consider (3.1) in which $f(x) = g(x) - h(x)$ is a DC function where $g, h\colon \mathbb{R}^d \to \mathbb{R}$ are convex functions. We also assume additionally that all constraint sets are convex and satisfy $\bigcap_{i=1}^q \operatorname{ri}(\Omega_i) \neq \emptyset$. By [6, Theorem 5.3], this condition ensures that

$$N(\bar{x}; \bigcap_{i=1}^q \Omega_i) = \sum_{i=1}^q N(\bar{x}; \Omega_i) \text{ for every } \bar{x} \in \bigcap_{i=1}^q \Omega_i. \tag{3.6}$$

Recall from [13] that an element $\bar{x} \in \mathbb{R}^d$ is a *critical point* of a DC function f with DC decomposition $f = g - h$ if $\partial g(\bar{x}) \cap \partial h(\bar{x}) \neq \emptyset$. Observe that (3.1) can be written as an unconstrained optimization problem using the indicator function as follows:

$$\min \quad \Big(g(x) + \delta(x, \bigcap_{i=1}^q \Omega_i)\Big) - h(x), x \in \mathbb{R}^d.$$

Define $v(x) = g(x) + \delta(x, \bigcap_{i=1}^q \Omega_i)$ for $x \in \mathbb{R}^d$. By (3.6),

$$\partial v(\bar{x}) = \partial g(\bar{x}) + N(\bar{x}; \bigcap_{i=1}^q \Omega_i) = \partial g(\bar{x}) + \sum_{i=1}^q N(\bar{x}; \Omega_i) \text{ for all } \bar{x} \in \mathbb{R}^d.$$

Thus, we call an element $\bar{x} \in \mathbb{R}^d$ a critical point of (3.1) if

$$\Big(\partial g(\bar{x}) + \sum_{i=1}^q N(\bar{x}; \Omega_i)\Big) \cap \partial h(\bar{x}) \neq \emptyset. \tag{3.7}$$

The objective function of (3.2) now becomes

$$f_\lambda(x) = g(x) + \frac{\lambda}{2} \sum_{i=1}^q d(x; \Omega_i)^2 - h(x).$$

Using Proposition 3.2, we have

$$f_\lambda(x) = \Big(g(x) + \frac{\lambda q}{2}\|x\|^2\Big) - \Big(h(x) + \frac{\lambda}{2} \sum_{i=1}^q \varphi_{\Omega_i}(x)\Big) = \tilde{g}_\lambda(x) - \tilde{h}_\lambda(x), \tag{3.8}$$

where $\tilde{g}_\lambda$ and $\tilde{h}_\lambda$ are functions defined on $\mathbb{R}^d$ by

$$\tilde{g}_\lambda(x) = g(x) + \frac{\lambda q}{2}\|x\|^2 \text{ and } \tilde{h}_\lambda(x) = h(x) + \frac{\lambda}{2} \sum_{i=1}^q \varphi_{\Omega_i}(x), x \in \mathbb{R}^d.$$

Proposition 3.4 *Suppose that* $\lim_{n\to\infty} \lambda_n = \infty$ *and* x_n *is a critical point of the DC function* $f_{\lambda_n} = \tilde{g}_{\lambda_n} - \tilde{h}_{\lambda_n}$ *given in* (3.8)*. Then every subsequential limit of the sequence* $\{x_n\}$ *is a critical point of* (3.1)*.*

Proof. Since x_n is a critical point of f_{λ_n} and by Proposition 3.3, there exist $v_n \in \partial g(x_n)$ and $w_n \in \partial h(x_n)$ such that

$$v_n + \lambda_n q x_n = w_n + \lambda_n \sum_{i=1}^q P(x_n; \Omega_i). \tag{3.9}$$

Let $\bar{x}$ be a subsequential limit of $\{x_n\}$. Without loss of generality, we can assume that $\{x_n\}$ converges to $\bar{x}$. Since any finite convex function is locally Lipschitz continuous, we can assume that both g and h are locally Lipschitz continuous around $\bar{x}$ with Lipschitz constant $L > 0$. Then

$$\|v_n\| \leq L \text{ and } \|w_n\| \leq L \text{ for sufficiently large } n. \tag{3.10}$$

By (3.9), (3.10) and the assumption that $\lambda_n \to \infty$ as $n \to \infty$,

$$\sum_{i=1}^{q} \Big(x_n - P(x_n; \Omega_i)\Big) = \frac{1}{\lambda_n}(w_n - v_n) \to 0 \text{ as } n \to \infty.$$

Letting $n \to \infty$ yields $\sum_{i=1}^{q} \Big(\bar{x} - P(\bar{x}; \Omega_i)\Big) = 0$, due to the continuity of projection operators onto convex sets. Note also that $\lambda_n \sum_{i=1}^{q} \Big(x_n - P(x_n; \Omega_i)\Big) \in \sum_{i=1}^{q} N(x_n; \Omega_i)$. This implies

$$w_n - v_n \in \sum_{i=1}^{q} N(x_n; \Omega_i) = N(x_n; \bigcap_{i=1}^{q} \Omega_i).$$

By (3.10), we can assume without loss of generality that $v_n \to \bar{v}$ and $w_n \to \bar{w}$ as $n \to \infty$. Then by passing to a limit, we have

$$\bar{w} - \bar{v} \in N(\bar{x}; \bigcap_{i=1}^{q} \Omega_i) = \sum_{i=1}^{q} N(\bar{x}; \Omega_i).$$

Observe also that $\bar{v} \in \partial g(\bar{x})$ and $\bar{w} \in \partial h(\bar{x})$. Therefore, (3.7) is satisfied and thus $\bar{x}$ is a critical point of (3.1). □

We continue by considering (3.4) in which

$$F(x^1, \ldots, x^k) = G(x^1, \ldots, x^k) - H(x^1, \ldots, x^k)$$

is a DC function, where $G, H \colon \mathbb{R}^{k \times d} \to \mathbb{R}$ are convex functions. From the proof of Corollary 3.2, we can rewrite (3.4) as

$$\begin{array}{ll} \min & F(\mathbf{X}) = G(\mathbf{X}) - H(\mathbf{X}) \\ \text{subject to} & \mathbf{X} \in \bigcap_{i=1}^{q} \mathbf{\Omega}_i. \end{array}$$

Recall that a point $\overline{\mathbf{X}} = (\bar{x}^1, \ldots, \bar{x}^k)$ is called a critical point of this problem if

$$\Big(\partial G(\overline{\mathbf{X}}) + \sum_{i=1}^{q} N(\overline{\mathbf{X}}; \mathbf{\Omega}_i)\Big) \cap \partial H(\overline{\mathbf{X}}) \neq \emptyset,$$

where $N(\overline{\mathbf{X}}; \mathbf{\Omega}_i) = N(\bar{x}^1; \Omega_i^1) \times \ldots \times N(\bar{x}^k; \Omega_i^k)$.

For $\mathbf{\Omega} \subset \mathbb{R}^{k \times d}$, based on Frobenious norm, we define

$$\varphi_{\mathbf{\Omega}}(\mathbf{X}) = \|\mathbf{X}\|_F^2 - d(\mathbf{X}; \mathbf{\Omega})^2 = 2 \sup \left\{ \langle \mathbf{X}, \mathbf{Y} \rangle - \frac{\|\mathbf{Y}\|_F^2}{2} \mid \mathbf{Y} \in \mathbf{\Omega} \right\}, \mathbf{X} \in \mathbb{R}^{k \times d}.$$

Observe that if $\mathbf{\Omega}_i = \Omega_i^1 \times \Omega_i^2 \ldots \times \Omega_i^k$ and $\mathbf{X} = (x^1, \ldots, x^k) \in \mathbb{R}^{k\times d}$, then

$$d(\mathbf{X}; \mathbf{\Omega}_i)^2 = \sum_{\ell=1}^{k} d(x^\ell; \Omega_i^\ell)^2 = \sum_{\ell=1}^{k} \Big(\|x^\ell\|^2 - \varphi_{\Omega_i^\ell}(x^\ell) \Big) = \|\mathbf{X}\|_F^2 - \sum_{\ell=1}^{k} \varphi_{\Omega_i^\ell}(x^\ell).$$

Therefore, $\varphi_{\mathbf{\Omega}_i}(\mathbf{X}) = \sum_{\ell=1}^{k} \varphi_{\Omega_i^\ell}(x^\ell)$.

In this new notation, the function F_λ in (3.5) can be rewritten as

$$F_\lambda(\mathbf{X}) = \Big(G(\mathbf{X}) + \frac{\lambda q}{2}\|\mathbf{X}\|_F^2 \Big) - \Big(H(\mathbf{X}) + \sum_{i=1}^{q} \varphi_{\mathbf{\Omega}_i}(\mathbf{X}) \Big) = G_1(\mathbf{X}) - H_1(\mathbf{X}),$$

where

$$G_1(\mathbf{X}) = G(\mathbf{X}) + \frac{\lambda q}{2}\|\mathbf{X}\|_F^2 \text{ and } H_1(\mathbf{X}) = H(\mathbf{X}) + \sum_{i=1}^{q} \varphi_{\mathbf{\Omega}_i}(\mathbf{X}), \mathbf{X} \in \mathbb{R}^{k\times d}.$$

We also recall that $\overline{\mathbf{X}} \in \mathbb{R}^{k\times d}$ is a critical point of (3.5) if

$$\partial G_1(\overline{\mathbf{X}}) \cap \partial H_1(\overline{\mathbf{X}}) \neq \emptyset.$$

The proof of the following result is similar to that of Proposition 3.4.

Proposition 3.5 *Suppose that* $\lim_{n\to\infty} \lambda_n = \infty$ *and* $\mathbf{X}_n = (x_n^1, \ldots, x_n^k) \in \mathbb{R}^{k\times d}$ *is a critical point of the function* F_{λ_n}. *Then every subsequential limit of* $\{\mathbf{X}_n\}$ *is a critical point of* (3.4).

4 Clustering with Constraints

In this section, we study problems of *clustering with constraints* in which the measurement distance is defined by the squared Euclidean norm. We seek k centers $x^1, \ldots, x^k \in \mathbb{R}^d$ of m data nodes $a^1, \ldots, a^m \in \mathbb{R}^d$ and impose the restriction that each $x^\ell \in \bigcap_{i=1}^{q} \Omega_i^\ell$ for some nonempty closed convex set $\Omega_i^\ell \subset \mathbb{R}^d$ with $\ell = 1, \ldots, k$ and $i = 1, \ldots, q$. Here, without loss of generality, we assume that the numbers of constraints for each center is equal to each other. The problem we are concerned with is given by

$$\begin{array}{ll} \min & \psi(x^1, \ldots, x^k) = \sum_{i=1}^{m} \min_{\ell=1,\ldots,k} \|x^\ell - a^i\|^2 \\ \text{subject to} & x^\ell \in \bigcap_{j=1}^{q} \Omega_j^\ell \text{ for } \ell = 1, \ldots, k. \end{array} \tag{4.1}$$

This problem can be converted to an unconstrained minimization problem:

$$\begin{array}{ll} \min & f(x^1, \ldots, x^k) = \frac{1}{2} \sum_{i=1}^{m} \min_{\ell=1,\ldots,k} \|x^\ell - a^i\|^2 + \frac{\tau}{2} \sum_{\ell=1}^{k} \sum_{i=1}^{q} [d(x^\ell; \Omega_i^\ell)]^2, \\ & x^1, \ldots, x^k \in \mathbb{R}^d, \end{array} \tag{4.2}$$

where $\tau > 0$ is a penalty parameter.

Recall from Proposition 3.3 that for any nonempty closed convex set Ω in $\mathbb{R}^d$,

$$[d(x; \Omega)]^2 = \|x\|^2 - \varphi_\Omega(x),$$

where $\varphi_\Omega(x) = 2\sup\left\{\langle x, w\rangle - \frac{1}{2}\|w\|^2 \mid w \in \Omega\right\}$ is a differentiable function with $\nabla\varphi_\Omega(x) = 2P(x;\Omega)$. Let us use the *minimum-sum principle* for k real numbers α_ℓ for $\ell = 1,\ldots,k$:

$$\min_{\ell=1,\ldots,k} \alpha_\ell = \sum_{\ell=1}^{k} \alpha_\ell - \max_{r=1,\ldots,k} \sum_{\ell=1,\ell\neq r}^{k} \alpha_\ell$$

to obtain a DC decomposition of f as follows

$$\begin{aligned} f(x^1,\ldots,x^k) = &\Big(\frac{1}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\|x^\ell - a^i\|^2 + \frac{\tau q}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2\Big) \\ &- \Big(\frac{1}{2}\sum_{i=1}^{m}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}(\|x^\ell - a^i\|)^2 + \frac{\tau}{2}\sum_{\ell=1}^{k}\sum_{i=1}^{q}\varphi_{\Omega_i^\ell}(x^\ell)\Big). \end{aligned}$$

We see that $f = g - h$ by defining

$$\begin{aligned} g_1(x^1,\ldots,x^k) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\|x^\ell - a^i\|^2,\ g_2(x^1,\ldots,x^k) = \frac{\tau q}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2\ , \\ h_1(x^1,\ldots,x^k) &= \frac{1}{2}\sum_{i=1}^{m}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}\|x^\ell - a^i\|^2,\ h_2(x^1,\ldots,x^k) = \frac{\tau}{2}\sum_{\ell=1}^{k}\sum_{i=1}^{q}\varphi_{\Omega_i^\ell}(x^\ell), \end{aligned}$$

and setting $g = g_1 + g_2$ and $h = h_1 + h_2$.

As discussed in the introduction, we may collect x^j into the variable matrix $\mathbf{X}$ and denote $\mathbf{\Omega}_i = \Omega_i^1 \times \Omega_i^2 \times \ldots \times \Omega_i^k \in \mathbb{R}^{k\times d}$ for $i = 1,\ldots,q$. Then (4.1) becomes

$$\min \psi(\mathbf{X}) \text{ subject to } \mathbf{X} \in \bigcap_{i=1}^{q} \mathbf{\Omega}_i.$$

We also collect a^i into the data matrix $\mathbf{A}$, and upon doing so we may express g in terms of the Frobenius norm, namely,

$$\begin{aligned} g_1(\mathbf{X}) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\|x^\ell - a^i\|^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\Big(\|x^\ell\|^2 - 2\langle x^\ell, a^i\rangle + \|a^i\|^2\Big) \\ &= \frac{m}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2 - \sum_{i=1}^{m}\sum_{\ell=1}^{k}\langle x^\ell, a^i\rangle + \frac{k}{2}\sum_{i=1}^{m}\|a^i\|^2 \\ &= \frac{m}{2}\|\mathbf{X}\|_F^2 - \langle \mathbf{X}, \mathbf{E}\mathbf{A}\rangle + \frac{k}{2}\|\mathbf{A}\|_F^2, \end{aligned}$$

where $\mathbf{E} \in \mathbb{R}^{k\times m}$ is the matrix of ones. In this form, it is easily seen that

$$\nabla g_1(\mathbf{X}) = m\mathbf{X} - \mathbf{E}\mathbf{A}.$$

Similarly, g_2 can be equivalently written as

$$g_2(\mathbf{X}) = \frac{\tau q}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2 = \frac{\tau q}{2}\big\|\mathbf{X}\big\|_F^2.$$

Hence, g_2 is differentiable and its gradient is given by $\nabla g_2(\mathbf{X}) = \tau q \mathbf{X}$. Therefore,

$$\nabla g(\mathbf{X}) = \nabla g_1(\mathbf{X}) + \nabla g_2(\mathbf{X}) = (m + \tau q)\mathbf{X} - \mathbf{EA}.$$

Based on the relation (2.2), finding $\mathbf{X} \in \partial g^*(\mathbf{Y})$ is equivalent to solving the equation

$$\mathbf{Y} = (m + \tau q)\mathbf{X} - \mathbf{EA}.$$

It follows that

$$\mathbf{X} = \frac{\mathbf{Y} + \mathbf{EA}}{m + \tau q} \in \partial g^*(\mathbf{Y}).$$

Our goal, then, is to find $\mathbf{Y}_p \in \partial h(\mathbf{X}_p)$ from which we will obtain $\mathbf{X}_{p+1}$ and thereby compute the first N terms of the sequence $\{\mathbf{X}_p\}$ via Algorithm 1. Toward this end we will find subgradients of the convex function h.

For each $i = 1, \ldots, m$, let $r(i) \in \{1, \ldots, k\}$ be an index for which

$$\sum_{\ell=1, \ell \neq r(i)}^{k} \|x^\ell - a^i\|^2 = \max_{r=1,\ldots,k} \sum_{\ell=1, \ell \neq r}^{k} \|x^\ell - a^i\|^2,$$

in which case we see that a subgradient $\mathbf{W} \in \partial h_1(\mathbf{X})$ is given by

$$\mathbf{W} = \sum_{i=1}^{m} \Big(\mathbf{X} - \mathbf{A}_i - e_{r(i)}(x^{r(i)} - a^i) \Big) = m\mathbf{X} - \mathbf{EA} - \sum_{i=1}^{m} e_{r(i)}(x^{r(i)} - a^i), \tag{4.3}$$

where $\mathbf{A}_i \in \mathbb{R}^{k \times d}$ is the matrix whose all rows are a^i and e_r is the $k \times 1$ column vector with a one in the r^{th} position and zeros elsewhere.

Now for $h_2(\mathbf{X}) = \frac{\tau}{2} \sum_{\ell=1}^{k} \sum_{i=1}^{q} \varphi_{\Omega_i^\ell}(x^\ell)$, we have

$$\frac{\partial h_2}{\partial x^j}(\mathbf{X}) = \frac{\tau}{2} \sum_{\ell=1}^{k} \sum_{i=1}^{q} \frac{\partial}{\partial x^j} \varphi_{\Omega_i^\ell}(x^\ell) = \tau \sum_{i=1}^{q} P(x^j; \Omega_i^j)$$

with $j = 1, \ldots, k$. Then $\mathbf{U} = \frac{1}{\tau} \nabla h_2(\mathbf{X})$ is the $k \times d$ matrix whose rows are $u^j = \sum_{i=1}^{q} P(x^j; \Omega_i^j)$.

The form of the DCA instructs us to find $\mathbf{Y}_p \in \partial h(\mathbf{X}_p)$ at the pth iteration, so we set $\mathbf{Y}_p = \mathbf{W} + \tau \mathbf{U}$. Combining the above results gives $\mathbf{X}_{p+1} = (\mathbf{W} + \tau \mathbf{U} + \mathbf{EA})/(m + \tau q)$. Substituting (4.3) for $\mathbf{W}$, we obtain the recursive relation

$$\mathbf{X}_{p+1} = \frac{1}{m + \tau q} \Big(m\mathbf{X}_p + \tau \mathbf{U} - \sum_{i=1}^{m} e_{r(i)} \big(x_p^{r(i)} - a^i \big) \Big),$$

where x_p^ℓ denotes the ℓ^{th} row of $\mathbf{X}_p$. The following algorithm summarizes the DCA-based procedure we just derived.

Inspecting (4.2), we see that for small τ our problem begins to resemble the associated unconstrained problem. For solving the clustering (4.1), we may gradually increase the value of the penalty parameter $\tau > 0$ by periodically multiplying by some $\sigma > 1$ and terminate whenever $\tau > \tau_f$. This may be accomplished by Algorithm 3. Notice that for the initial choice of τ, the maximum number of overall iterations of Algorithm 3 is $N\lceil \log_\sigma(\tau_f/\tau) \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function.

Algorithm 2 : DC program for (4.2)

INPUT: $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, N, \tau$

for $p = 1, \ldots, N$ **do**

  for $i = 1, \ldots, m$ **do**

    Find $r(i)$ s.t. $\|x_{p-1}^{r(i)} - a^i\|^2 = \min\{\left\|x_{p-1}^\ell - a^i\right\|^2 \mid \ell = 1, \ldots, k\}$

    Set $\mathbf{W}_i := e_{r(i)}(x_{p-1}^{r(i)} - a^i)$

  end

  for $\ell = 1, \ldots, k$ **do**

    Find $u^\ell := \sum_{j=1}^q P(x_{p-1}^\ell; \Omega_j^\ell)$

  end

  Set $\mathbf{X}_p := \frac{1}{m+\tau q}\Big(m\mathbf{X}_{p-1} + \tau\mathbf{U} - \sum_{i=1}^m \mathbf{W}_i\Big)$

end

OUTPUT: $\mathbf{X}_N$

Algorithm 3 : Penalty DC program for (4.1)

INPUT: $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, N, \tau, \sigma, \tau_f$

while $\tau < \tau_f$ **do**

  Find $\mathbf{X}_N$ by executing **Algorithm 2** with $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, N, \tau$

  Reassign $\mathbf{X}_0 := \mathbf{X}_N$

  Reassign $\tau := \sigma\tau$

end

OUTPUT: $\mathbf{X}_N$

5 Set Clustering with Constraints

In this section, we turn our attention to a model of *set clustering* with constraints, i.e., for given m subsets $\Lambda_1, \ldots, \Lambda_m \subset \mathbb{R}^d$, we seek k cluster centers $x^\ell \in \bigcap_{j=1}^q \Omega_j^\ell$ for $\ell = 1, \ldots, k$, where each Ω_j^ℓ is a subset of $\mathbb{R}^d$. The measurement distance is defined by the squared distance functions to the sets involved. The optimization modeling of the problem to be solved is given by

$$\begin{array}{ll} \min & \psi(x^1, \ldots, x^k) = \sum_{i=1}^m \min_{\ell=1,\ldots,k}[d(x^\ell; \Lambda_i)]^2 \\ \text{subject to} & x^\ell \in \bigcap_{j=1}^q \Omega_j^\ell \text{ for } \ell = 1, \ldots, k. \end{array} \tag{5.1}$$

Throughout this section, we assume that Λ_i for $i = 1, \ldots, m$ and Ω_j^ℓ for $j = 1, \ldots, q$ and $\ell = 1, \ldots, k$ are nonempty, closed and convex.

Using the penalty method based on distance functions with a parameter $\tau > 0$, we consider the constrained set clustering model:

$$\begin{array}{ll} \min & f(x^1, \ldots, x^k) = \frac{1}{2}\sum_{i=1}^m \min\limits_{\ell=1,\ldots,k}[d(x^\ell; \Lambda_i)]^2 + \frac{\tau}{2}\sum_{\ell=1}^k \sum_{j=1}^q [d(x^\ell; \Omega_j^\ell)]^2, \\ & x^1, \ldots, x^k \in \mathbb{R}^d. \end{array} \tag{5.2}$$

We will now find a DC decomposition of $f = g - h$ as follows. For each $i = 1, \ldots, m$, we have

$$\begin{aligned}\min_{\ell=1,\ldots,k}[d(x^\ell;\Lambda_i)]^2 &= \sum_{\ell=1}^{k}[d(x^\ell;\Lambda_i)]^2 - \max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2\\ &= \sum_{\ell=1}^{k}\Big(\|x^\ell\|^2 - \varphi_{\Lambda_i}(x^\ell)\Big) - \max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2\\ &= \|\mathbf{X}\|_F^2 - \Big(\sum_{\ell=1}^{k}\varphi_{\Lambda_i}(x^\ell) + \max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2\Big).\end{aligned}$$

Furthermore, we have

$$\sum_{\ell=1}^{k}\sum_{j=1}^{q}[d(x^\ell;\Omega_j^\ell)]^2 = \sum_{\ell=1}^{k}\sum_{j=1}^{q}\Big(\|x^\ell\|^2 - \varphi_{\Omega_j^\ell}(x^\ell)\Big) = q\,\|\mathbf{X}\|_F^2 - \sum_{\ell=1}^{k}\sum_{j=1}^{q}\varphi_{\Omega_j^\ell}(x^\ell).$$

Let

$$\begin{aligned}&g_1(\mathbf{X}) = \frac{m}{2}\|\mathbf{X}\|_F^2,\ g_2(\mathbf{X}) = \frac{\tau q}{2}\,\|\mathbf{X}\|_F^2\,,\\ &h_1(\mathbf{X}) = \sum_{i=1}^{m}\Big(\frac{1}{2}\sum_{\ell=1}^{k}\varphi_{\Lambda_i}(x^\ell) + \frac{1}{2}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2\Big),\ h_2(\mathbf{X}) = \frac{\tau}{2}\sum_{\ell=1}^{k}\sum_{j=1}^{q}\varphi_{\Omega_j^\ell}(x^\ell),\end{aligned}$$

in which case we have the DC decomposition $f = g - h$, where $g = g_1 + g_2$ and $h = h_1 + h_2$ are convex.

Using the relation (2.2), we can easily see that $\mathbf{X} = \frac{1}{m+\tau q}\mathbf{Y} \in \partial g^*(\mathbf{Y})$. To apply the DCA from Algorithm 1, we also need to find $\mathbf{Y} \in \partial h(\mathbf{X})$ as $\mathbf{Y} = \mathbf{V} + \mathbf{U}$, where $\mathbf{V} \in \partial h_1(\mathbf{X})$ and $\mathbf{U} \in \partial h_2(\mathbf{X})$.

Now, we focus on finding $\mathbf{V} \in \partial h_1(\mathbf{X})$. Define

$$D_i(\mathbf{X}) = \frac{1}{2}\sum_{\ell=1}^{k}\varphi_{\Lambda_i}(x^\ell),$$

and

$$F_i(\mathbf{X}) = \frac{1}{2}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2,\ i = 1,\ldots,m.$$

Then $h_1(\mathbf{X}) = \sum_{i=1}^{m}[D_i(\mathbf{X}) + F_i(\mathbf{X})]$. Based on Proposition 3.3, we see that $\nabla D_i(\mathbf{X})$ is the $k \times d$ matrix given by

$$\nabla D_i(\mathbf{X}) = \begin{bmatrix} P(x^1;\Lambda_i) \\ \vdots \\ P(x^k;\Lambda_i)\end{bmatrix}.$$

For each $i = 1, \ldots, m$, choose an index $r(i)$ such that

$$\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}[d(x^\ell;\Lambda_i)]^2 = \sum_{\ell=1,\ell\neq r(i)}^{k}[d(x^\ell;\Lambda_i)]^2.$$

Now, for $j = 1, \ldots, k$, define

$$v_i^j = \begin{cases} x^j - P(x^j; \Lambda_i) & \text{if } j \neq r(i), \\ 0 & \text{if } j = r(i). \end{cases}$$

By (2.1) and the fact that $\nabla[d(x;\Lambda)]^2 = 2\Big(x - P(x;\Lambda)\Big)$ for a nonempty closed convex set Λ, the matrix V_i whose j^{th} row is v_i^j defines a subgradient of F_i at $\mathbf{X}$. It follows that such a subgradient $\mathbf{V}$ is

$$\mathbf{V} = m\mathbf{X} - \sum_{i=1}^{m} e_{r(i)}\Big(x^{r(i)} - P(x^{r(i)}; \Lambda_i)\Big).$$

As computed in the previous section, $\nabla h_2(\mathbf{X}) = \tau\mathbf{U}$, where $\mathbf{U}$ is the $k \times d$ matrix whose ℓ^{th} row is $\sum_{j=1}^{k} P(x^\ell; \Omega_j^\ell)$ for $\ell = 1, \ldots, k$. Consequently, the $k \times d$ matrix

$$\mathbf{Y} = m\mathbf{X} - \sum_{i=1}^{m} e_{r(i)}\Big(x^{r(i)} - P(x^{r(i)}; \Lambda_i)\Big) + \tau\mathbf{U} = m\mathbf{X} + \tau\mathbf{U} - \sum_{i=1}^{m} e_{r(i)}\Big(x^{r(i)} - P(x^{r(i)}; \Lambda_i)\Big)$$

belongs to $\partial h(\mathbf{X})$.

Now, for $p \in I\!N$ such that $\mathbf{X}_{p-1}$ is given, one has

$$\mathbf{Y}_{p-1} = m\mathbf{X}_{p-1} + \tau\mathbf{U}_p - \sum_{i=1}^{m} e_{r(i)}\Big(x_{p-1}^{r(i)} - P(x_{p-1}^{r(i)}; \Lambda_i)\Big) \in \partial H(\mathbf{X}_{p-1}),$$

where x_p^ℓ is the ℓ^{th} row of $\mathbf{X}_p$ and $\mathbf{U}_p$ is the $k \times d$ matrix whose ℓ^{th} row is $\sum_{j=1}^{k} P(x_p^\ell; \Omega_j^\ell)$ for $l = 1, \ldots, k$. It follows that $\mathbf{X}_p$ from the DCA in Algorithm 1 can be determined by

$$\mathbf{X}_p = \frac{1}{\tau q + m}\Big(m\mathbf{X}_{p-1} + \tau\mathbf{U}_p - \sum_{i=1}^{m} e_{r(i)}\big(x_{p-1}^{r(i)} - P(x_{p-1}^{r(i)}; \Lambda_i)\big)\Big).$$

We now adapt Algorithm 4 to solve our set clustering problem. Just as in the previous section, we gradually increase the value of the penalty parameter $\tau > 0$ by periodically multiplying it by some $\sigma > 1$ and stopping when $\tau > \tau_f > 0$. This may be accomplished by Algorithm 5. We again see that for an initial choice of $\tau = \tau_0$, the maximum number of overall iterations of Algorithm 5 is $N\lceil \log_\sigma(\tau_f/\tau_0)\rceil$.

Algorithm 4 : DC program for (5.2)

INPUT: $\mathbf{X}_0, \Lambda_i, \{\Omega_j^\ell\}_{j=1,\dots,q}^{\ell=1,\dots,k}, N, \tau$

for $p = 1, \dots, N$ **do**
  for $i = 1, \dots, m$ **do**
    for $\ell = 1, \dots, k$ **do**
      Set $w_i^\ell := P(x_{p-1}^\ell; \Lambda_i)$
    end
    Find $r(i)$ s.t. $\|x_{p-1}^{r(i)} - w_i^{r(i)}\|^2 = \min\limits_{\ell=1,\dots,k} \|x_{p-1}^\ell - w_i^\ell\|^2$
  end
  for $\ell = 1, \dots, k$ **do**
    Find $u^\ell := \sum_{j=1}^q P(x_{p-1}^\ell; \Omega_j^\ell)$
  end
  $\mathbf{X}_p := \frac{1}{\tau q + m}\Big(m\mathbf{X}_{p-1} + \tau \mathbf{U}_p - \sum_{i=1}^m e_{r(i)}\big(x_{p-1}^{r(i)} - w_i^{r(i)}\big)\Big)$
end

OUTPUT: $\mathbf{X}_N$

Algorithm 5 : Penalty DC program for (5.1)

INPUT: $\mathbf{X}_0, \{\Lambda_i\}_{i=1}^m, \{\Omega_j^\ell\}_{j=1,\dots,q}^{\ell=1,\dots,k}, N, \tau, \tau_f, \sigma$

while $\tau < \tau_f$ **do**
  Find $\mathbf{X}_N$ by executing **Algorithm 4** with $\mathbf{X}_0, \{\Lambda_i\}_{i=1}^m, \{\Omega_j^\ell\}_{j=1,\dots,q}^{\ell=1,\dots,k}, \tau, N$
  Reassign $\mathbf{X}_0 := \mathbf{X}_N$
  Reassign $\tau := \sigma\tau$
end

OUTPUT: $\mathbf{X}_N$

6 Multifacility Location with Constraints

Given a set of m points (nodes) $a^1, a^2, \dots, a^m$ in $\mathbb{R}^d$, our goal is find k centers x^ℓ for $\ell = 1, \dots, k$, which must be in constraint sets $\bigcap_{i=1}^q \Omega_i^\ell$ for $l = 1, .., k$, such that the *transportation cost* to the nodes is minimized. The same setting in Section 4 gives us the constrained minimization problem:

$$\min \psi(\mathbf{X}) \text{ subject to } \mathbf{X} \in \bigcap_{i=1}^q \mathbf{\Omega}_i, \tag{6.1}$$

where the total cost now is given by

$$\psi(\mathbf{X}) = \psi(x^1, \dots, x^k) = \sum_{i=1}^m \min_{\ell=1,\dots,k} \|x^\ell - a^i\|.$$

This problem can be converted to an unconstrained minimization problem:

$$\begin{aligned} \min \quad & f_\tau(x^1, \dots, x^k) = \textstyle\sum_{i=1}^m \min_{\ell=1,\dots,k} \|x^\ell - a^i\| + \frac{\tau}{2} \sum_{\ell=1}^k \sum_{i=1}^q [d(x^\ell; \Omega_i^\ell)]^2, \\ & x^1, \dots, x^k \in \mathbb{R}^d. \end{aligned} \tag{6.2}$$

where $\tau > 0$ is a parameter.

We apply Nesterov's smoothing techniques from [8] to approximate the objective function f_τ by a new DC function which is favorable for applying the DCA.

$$f_{\tau,\mu}(x^1,\ldots,x^k) = \Big(\frac{\mu}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\left\|\frac{x^\ell - a^i}{\mu}\right\|^2 + \frac{\tau q}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2\Big) - \Big(\frac{\mu}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\left[d\left(\frac{x^\ell - a^i}{\mu};\mathbb{B}\right)\right]^2 + \sum_{i=1}^{m}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}\|x^\ell - a^i\| + \frac{\tau}{2}\sum_{\ell=1}^{k}\sum_{i=1}^{q}\varphi_{\Omega_i^\ell}(x^\ell)\Big).$$

In what follows, we use f instead of $f_{\tau,\mu}$ for the simplicity of notations. The original clustering problem now can be solved using a DC programming:

$$\min f(x^1,\ldots,x^k) = g(x^1,\ldots,x^k) - h(x^1,\ldots,x^k),\ \ x^1,\ldots,x^k \in \mathbb{R}^d.$$

In this formulation, g and h are convex functions on $(\mathbb{R}^d)^k$ defined by

$$g(x^1,\ldots,x^k) = g_1(x^1,\ldots,x^k) + g_2(x^1,\ldots,x^k),$$
$$h(x^1,\ldots,x^k) = h_1(x^1,\ldots,x^k) + h_2(x^1,\ldots,x^k) + h_3(x^1,\ldots,x^k),$$

with their respective components defined as

$$g_1 = \frac{\mu}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\left\|\frac{x^\ell - a^i}{\mu}\right\|^2,\ g_2 = \frac{\tau q}{2}\sum_{\ell=1}^{k}\|x^\ell\|^2,$$
$$h_1 = \frac{\mu}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\left[d\left(\frac{x^\ell - a^i}{\mu};\mathbb{B}\right)\right]^2,\ h_2 = \sum_{i=1}^{m}\max_{r=1,\ldots,k}\sum_{\ell=1,\ell\neq r}^{k}\|x^\ell - a^i\|,\ h_3 = \frac{\tau}{2}\sum_{\ell=1}^{k}\sum_{i=1}^{q}\varphi_{\Omega_i^\ell}(x^\ell).$$

The function g_1 can be equivalently written as

$$\begin{aligned} g_1(\mathbf{X}) &= \frac{1}{2\mu}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\|x^\ell - a^i\|^2 \\ &= \frac{1}{2\mu}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\Big(\|x^\ell\|^2 - 2\langle x^\ell, a^i\rangle + \|a^i\|^2\Big) \\ &= \frac{1}{2\mu}\Big(m\big\|\mathbf{X}\big\|_F^2 - 2\Big\langle\mathbf{X},\mathbf{EA}\Big\rangle + k\big\|\mathbf{A}\big\|_F^2\Big). \end{aligned}$$

Note that g_1 is differentiable and its gradient is given by

$$\nabla g_1(\mathbf{X}) = \frac{1}{\mu}\left[m\mathbf{X} - \mathbf{EA}\right].$$

The function g_2 is the same as before so its gradient is given by

$$\nabla g_2(\mathbf{X}) = \tau q\mathbf{X}.$$

Since $g(\mathbf{X}) = g_1(\mathbf{X}) + g_2(\mathbf{X})$, its gradient can be computed by

$$\begin{aligned}\nabla g(\mathbf{X}) &= \nabla g_1(\mathbf{X}) + \nabla g_2(\mathbf{X}) \\ &= \frac{1}{\mu}\Big(m\mathbf{X} - \mathbf{EA}\Big) + \tau q\mathbf{X} \\ &= (\frac{m}{\mu} + \tau q)\mathbf{X} - \frac{1}{\mu}\mathbf{S},\end{aligned}$$

where $\mathbf{S} = \mathbf{EA}$. The latter can equivalently be written as

$$\mathbf{Y} = (\frac{m}{\mu} + \tau q)\mathbf{X} - \frac{1}{\mu}\mathbf{S}.$$

Our goal now is to compute $\nabla g^*(\mathbf{Y})$, which can be accomplished by the relation (2.2). Then with some algebraic manipulations, we can show that

$$\nabla g^*(\mathbf{Y}) = \mathbf{X} = \frac{\mu\mathbf{Y} + \mathbf{S}}{m + \mu\tau q}.$$

Next, we will demonstrate in more details the techniques we used in finding a subgradient for the convex function h. Recall that h is defined by

$$h(\mathbf{X}) = \sum_{i=1}^{3} h_i(\mathbf{X}).$$

We will start with the function h_1 given by

$$h_1(\mathbf{X}) \;=\; \frac{\mu}{2}\sum_{i=1}^{m}\sum_{\ell=1}^{k}\left[d\left(\frac{x^\ell - a^i}{\mu}; \mathbb{B}\right)\right]^2.$$

Similar to the situation in [8], we get

$$\frac{\partial h_1}{\partial x^\ell}(\mathbf{X}) \;=\; \sum_{i=1}^{m}\left(\frac{x^\ell - a^i}{\mu} - P\Big(\frac{x^\ell - a^i}{\mu}; I\!B\Big)\right).$$

Thus, for $\ell = 1, 2, \ldots, k$, $\nabla h_1(\mathbf{X}) = \mathbf{Z}$ is the $k \times d$ matrix whose ℓ^{th} row is $\frac{\partial h_1}{\partial x^\ell}(\mathbf{X})$.

Let us compute a subgradient of h_2 as in [8]

$$h_2(\mathbf{X}) \;=\; \sum_{i=1}^{m}\max_{\ell=1,\ldots,k}\sum_{j=1, j\neq\ell}^{k}\|x^j - a^i\| = \sum_{i=1}^{m}\gamma_i(\mathbf{X}),$$

where $\gamma_i(\mathbf{X}) = \max_{\ell=1,\ldots,k}\sum_{j=1,j\neq\ell}^{k}\|x^j - a^i\|$. For each $i = 1, \ldots, m$, define

$$\gamma_{i\ell}(\mathbf{X}) = \sum_{j=1, j\neq\ell}^{k}\|x^j - a^i\|, \ell = 1, \ldots, k.$$

Then $\gamma_i(\mathbf{X}) = \max_{\ell=1,\ldots,k}\gamma_{i\ell}(\mathbf{X})$.

Based on the subdifferential formula for maximum functions, for each $i = 1, \ldots, m$, we find $\mathbf{W}_i \in \partial\gamma_i(\mathbf{X})$. Then define $\mathbf{W} = \sum_{i=1}^{m} \mathbf{W}_i$ to get a subgradient of the function h_2 at $\mathbf{X}$ by the subdifferential sum rule. To accomplish this goal, we first choose an index $\ell^* = 1, \ldots, k$ such that $\gamma_i(\mathbf{X}) = \gamma_{i\ell^*}(\mathbf{X}) = \sum_{j=1, j\neq\ell^*}^{k} \|x^j - a^i\|$. Using the familiar subdifferential formula of the Euclidean norm function, the j^{th} row w_i^j for $j \neq \ell^*$ of the matrix $\mathbf{W}_i$ is determined as follows

$$w_i^j = \begin{cases} \frac{x^j - a^i}{\|x^j - a^i\|} & \text{if } x^j \neq a^i, \\ 0 & \text{if } x^j = a^i. \end{cases}$$

The ℓ^{*th} row of the matrix $\mathbf{W}_i$ is $w_i^{\ell^*} = 0$.

The procedure for computing $\partial h_3(\mathbf{X})$ is the same in Section 4. Let $\mathbf{U}$ be the matrix whose rows are $\sum_{i=1}^{q} P(x^\ell; \Omega_i^\ell)$, for $\ell = 1, \ldots, k$, then $\nabla h_3(\mathbf{X}) = \tau\mathbf{U}$.

At this point, we are ready to give a new DCA-based algorithm for our problem.

Algorithm 6 : DC program for (6.2)

INPUT: $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, \tau, \mu, N \in \mathbb{N}$.
for $p = 1, 2, \cdots, N$ **do**
  Find $\mathbf{Y}_p := \mathbf{Z}_p + \mathbf{W}_p + \tau\mathbf{U}_p$ where
    $\mathbf{Z}_p := \nabla h_1(\mathbf{X}_p)$
    $\mathbf{W}_p \in \partial h_2(\mathbf{X}_p)$
    $\mathbf{U}_p := \nabla h_3(\mathbf{X}_p)$
  Find $\mathbf{X}_{p+1} := \frac{\mu(\mathbf{Z}_p + \mathbf{W}_p + \tau\mathbf{U}_p) + \mathbf{S}}{m + \mu\tau q}$
end
OUTPUT: $\mathbf{X}_N$

We also present below an adapted version of Algorithm 6 for solving (6.1). We may improve Algorithm 6 by gradually increasing and decreasing the value of the penalty parameter τ and the smoothing parameter μ respectively. This can be done by periodically multiplying them by some $\sigma > 1$, $0 < \delta < 1$ and stopping when $\tau > \tau_f$, $\mu < \mu_f$.

Algorithm 7 : Penalty DC program for (6.1)

INPUT: $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, N, \tau, \sigma, \tau_f, \mu, \delta, \mu_f$
while $\tau < \tau_f$ and $\mu > \mu_f$ **do**
  Find $\mathbf{X}_N$ by executing **Algorithm 6** with $\mathbf{A}, \mathbf{X}_0, \{\Omega_j^\ell\}_{j=1,\ldots,q}^{\ell=1,\ldots,k}, N, \tau, \mu$
  Reassign $\mathbf{X}_0 := \mathbf{X}_N$
  Reassign $\tau := \sigma\tau$
  Reassign $\mu := \delta\mu$
end
OUTPUT: $\mathbf{X}_N$

7 Numerical Experiments

We now implement proposed algorithms to solve some constrained clustering and multifacility location problems in a number of examples. All the test are implemented in MATLAB. Instead of choosing the number of iterations N in advance, we terminate the DCA in Algorithms 2, 4 and 6 whenever $\|\mathbf{X}_{p+1} - \mathbf{X}_p\|_F < 10^{-8}$. In all examples, we initialize starting centers $\mathbf{X}_0$ as an $k \times d$ matrix whose all rows are the mean of the dataset $\mathbf{A}$.

7.1 Constrained Clustering

Example 7.1 We now consider the dataset EIL76 taken from the Traveling Salesman Problem Library [10]. We impose the following constraints on the solution:

1. The first center is a common point of a box whose vertices are $(40, 40)$; $(40, 60)$; $(20, 60)$; $(20, 40)$ and a ball of radius $r = 7$ centered at $(20, 60)$.
2. The second center is in the intersection of two balls of the same radius $r = 7$, centered at $(35, 20)$ and $(45, 22)$, respectively.

Choosing $\tau = 1$, $\sigma = 10$, $\tau_f = 10^8$, Algorithm 3 yields an approximate solution: $\mathbf{X} = \begin{pmatrix} 26.69959 & 57.97125 \\ 41.06910 & 23.48799 \end{pmatrix}$, with the cost $\psi(\mathbf{X}) = 33576.25387$; see Figure 1.

7.2 Set Clustering with Constraints

Example 7.2 We now use Algorithm 5 to solve a set clustering problem with constraints. We consider the latitude and longitude of the 50 most populous US cities taken from 2014 United States Census Bureau data [5], and approximate each city by a ball with radius $0.1\sqrt{\frac{A}{\pi}}$ where A is the city's reported area in square miles.

We use Algorithm 5 for solving 3-center problem generated by this 50-set dataset with requirement that each center must belong to the intersection of two balls. The centers of these constrained balls are the columns of the matrix below

$$\begin{pmatrix} -80 & -80 & -92 & -90 & -115 & -110 \\ 34 & 38 & 37 & 40 & 45 & 40 \end{pmatrix}$$

with corresponding radii given by $\begin{pmatrix} 2 & 3 & 4 & 3 & 4 & 4 \end{pmatrix}$. The result is plotted in Figure 2 using a plate Carrée projection [6].

We again choose $\tau = 1$, $\sigma = 10$, $\tau_f = 10^8$, Algorithm 5 yields an approximate optimal value

[5] https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population

[6] https://www.mathworks.com/help/map/pcarree.html

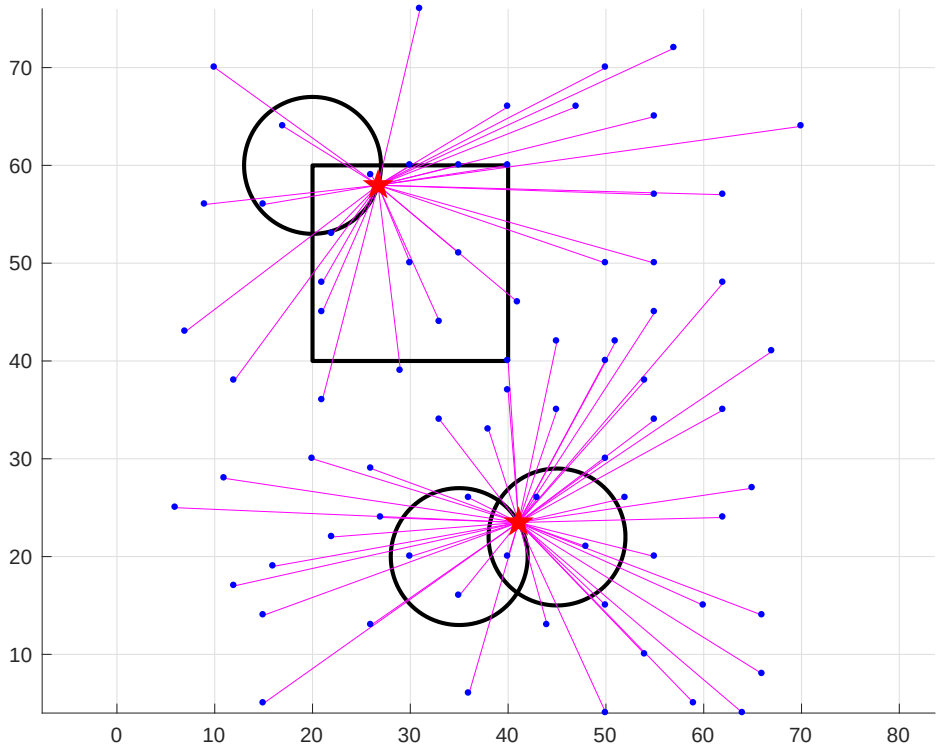

Figure 1: A 2-center constrained clustering problem for dataset EIL76.

$\psi(\mathbf{X}) = 2271.09657$ at an aproximate solution given by

$$\mathbf{X} = \begin{pmatrix} -79.32172 & 35.88148 \\ -91.93134 & 37.70436 \\ -113.82289 & 41.17711 \end{pmatrix}.$$

7.3 Multifacility Location with Constraints

Example 7.3 We now test Algorithm 7 on a data set $\mathbf{A}$ containing random points in 4 balls of radius $r = 0.3$ centered at $(2, 2)$, $(4, 2)$, $(4, 4)$ and $(2, 4)$. Let $k = 4$ and the constraint be the ball with the same radius, centered at $(3, 3)$. We use the *kmeans* (a MATLAB built in function) to partition the nodes into 4 clusters first, and then we selected the 4 cluster centroid locations as starting centers. We choose $\tau = 1$, $\sigma = 10$, $\tau_f = 10^8$, $\mu = 1$, $\delta = 0.75$, $\mu_f = 10^{-6}$. Typical centers are the intersections of the constraint ball boundary and the line connecting centers of each ball to the center of the constraint one. A visualization is shown in Figure 3.

Example 7.4 Next we consider the latitude and longitude data of the $m = 988$ most-populated cities in the contiguous 48 United States [14]. We impose the following constraints on the solution:

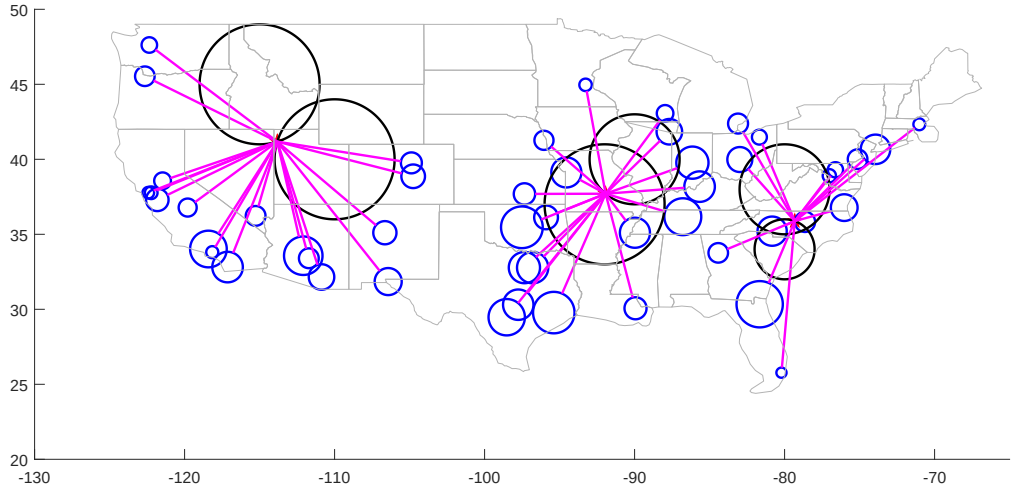

Figure 2: A 3-center set clustering problems with 50 most populous US cities. Each city is approximated by a ball proportional to its area.

1. One center is to lie west of $-115°$ longitude and within $4°$ latitude/longitude of Caldwell, Idaho.
2. One center is to lie within the state of Colorado and within $6°$ latitude/longitude of Oklahoma City, Oklahoma.
3. One center is to lie within $2°$ latitude/longitude of Skokie, Illinois and the triangle with vertices at Cleveland, Ohio; Atlanta, Georgia; and Des Moine, Iowa.
4. One center is to lie within $4°$ latitude/longitude of New York, NY and Washington,DC.

Employing Algorithms 6 and 7 with $\tau = 1$, $\sigma = 100$, $\tau_f = 10^8$, $\mu = 1$, $\delta = 0.85$, $\mu_f = 10^{-6}$, we terminate when $\|\mathbf{X}_{p+1} - \mathbf{X}_p\|_F < 10^{-6}$ and find final centers at

$$\mathbf{X} = \begin{pmatrix} -118.03185 & 39.89550 \\ -102.04996 & 36.99996 \\ -87.93854 & 40.90443 \\ -76.63980 & 38.67968 \end{pmatrix}$$

with an objective value $\psi(\mathbf{X}) = 42586.65060$; see Figure 4.

References

[1] L. T. H. An, M. T. Belghiti, P. D. Tao, A new efficient algorithm based on DC programming and DCA for clustering, J. Glob. Optim .**27**, (2007), 503–608.

[2] L. T. H. An, L. H. Minh, P. D. Tao, New and efficient DCA based algorithms for minimum sum-of-squares clustering, Pattern Recognition .**47** (2014), 388–401.

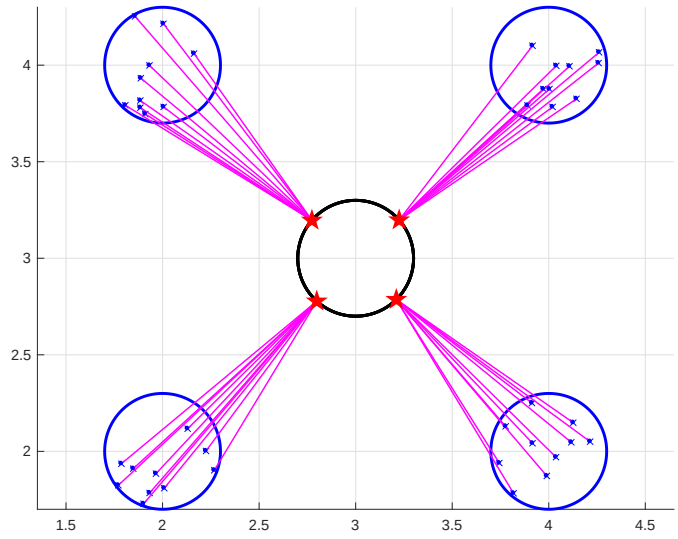

Figure 3: A 4-center multifacility location with one ball constraint

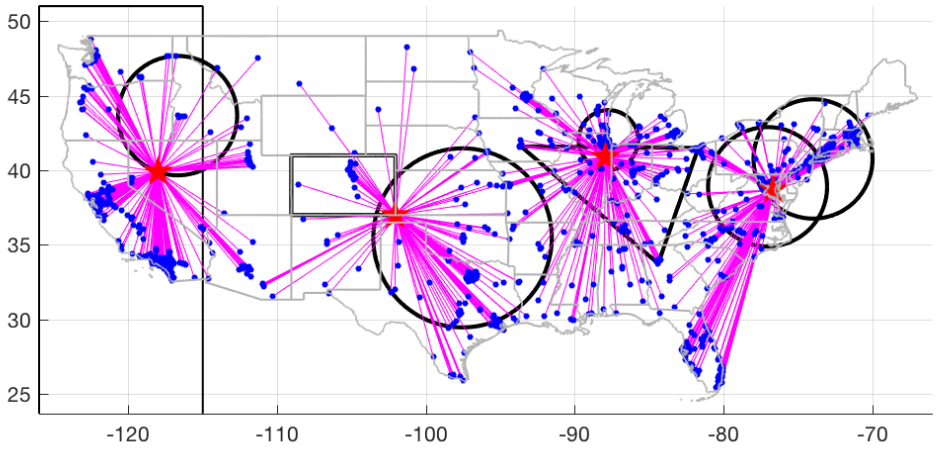

Figure 4: A 4-center constrained multifacility location problem with US cities dataset.

[3] E. Chi, H. Zhou, K. Lange, Distance majorization and its applications, Math. Program. Series A .**146**, (2014), 409–436.

[4] J. B. Hiriart-Urruty and C. Lemaréchal, *Funndamental of Convex Analysis*, Springer-Verlag, 2001.

[5] B. S. Mordukhovich and N. M. Nam, *An Easy Path to Convex Analysis and Applications*, Morgan & Claypool Publishers, San Rafael, CA, 2014.

[6] B. S. Mordukhovich and N. M. Nam, Geometric approach to convex subdifferential calculus, Optim. **66**, (2017), 839–873.

[7] N. M. Nam, W. Geremew, S. Reynolds, T. Tran, The Nesterov Smoothing Technique and Minimizing Differences of Convex Functions for Hierarchical Clustering, submitted.

[8] N. M. Nam, R. B. Rector, D. Giles, Minimizing Differences of Convex Functions with Applications to Facility Location and Clustering, submitted.

[9] J. Nocedal, S.J. Wright, *Numerical Optimization* , Springer, New York, 2nd Edition, 2006.

[10] G. Reinelt, TSPLIB: A Traveling Salesman Problem Library, *ORSA Journal of Computing*. **3**, 376384, 1991.

[11] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[12] P. D. Tao, L. T. H. An, Convex analysis approach to D.C. programming: Theory, algorithms and applications, Acta Math. Vietnam. **22** (1997), 289–355.

[13] P.D. Tao, L. T. H. An, A D.C. optimization algorithm for solving the trust-region subproblem, SIAM J. Optim. **8** (1998), 476–505.

[14] United States Cities Database. *Simple Maps: Geographic Data Products, 2017*, `http://simplemaps.com/data/us-cities`.